**American International University- Bangladesh**

DATABASE PROJECT ON SUPER SHOP

Farhan Rahman Arnob

ID#13-24160-2

Sumaya Jahan

ID#13-24607-2

INDEX

|  |  |
| --- | --- |
| Table of Contents | Page |
| 1.Business and System Summery | 3 |
| 2. Overview of Business Environment and Project Objectives | 3-4 |
| 3. Summarization of the Developed Database | 4 |
| 4. Justification from my side in the database | 4 |
| 5. An ERD Diagram in the original | 5 |
| 6.Normalization up to 3rd Normal form | 5-6 |
| 7. A relationship diagram after normalization | 6 |
| 8.Creating table | 6-13 |
| 9.Screenshot of Sample data of Table | 13-18 |
| 10.Sample and complex views | 19-20 |
| 11.Screen shot of all over database | 21 |

***Business And System Summary :-***

A to Z point is established in 2010 to make traveling fun and comfortable for the all classes of people of home and abroad. The head office of the company is situated at gulshan, Dhaka. There are many branches of the company around Dhaka city. Every branches has a manager with sufficient employees.

A to z point provides various kind of products for customers. Customers can come for shopping for any products that is offered by the shop. As the shop system, any branch of A to Z point is to take customers orders for Products. According to the customers shoppings a bill slip is generated automatically which is checked by agency manager later and sent to the belonging customer. As far the shop system customer pay his/her payment using the billing slip. Customer’s payment is confirmed by manager if the payment is ok then the shopping has been confirmed for the customer.

From this discussion, we have identified the data entities required in the data model for the A to Z point: customers,shopping,products,bill,manager

***Overview of Business Environment and Project Objectives:-***

A to Z point offers different productss. Each products has different product\_id and producte\_type. Customers come for shopping including products. Products have products\_price. This price generate the bill. In pay slip there may be discount. And Finally the Net\_total\_amount. Customers pay their payment through billing\_id. Billslip cannot be generate without shopping, and shopping cannot be generate without products.

Concerning the Customers:

1. A customer has an unique customer\_id and a specific contact\_number
2. A customers can request for one or many shopping. And many customers can come for many shoppings.

Concerning the shopping:

1. We record each shopping of client
2. Each shopping corresponds to a product
3. A shopping cannot exists without a product
4. And over time there will be many-to-one products for shopping

Concerning Employees:

1. Every employee has an unique employee\_id
2. Every employee is managed by a manager

Concerning Manager:

1. Manager manage all salesman
2. Manager manage the shop.

Concerning products:

1. A products has an unique product\_id and specific product-type and price
2. Shop has different products.

Concerning Bill:

1. A bill has an unique bill\_id
2. A booking generate one billslip.

***Summerization of the Developed Database:-***

The database is developed thinking the both sides of the business shop and customers. Using this database a customer can choose products and can come for shopping. On the other other hand shop can operate it’s operation’s on on the basis of various aspect.

***Justification from my side in the database:-***

I think the database meets the business policy of A toZ point According to the business policy what the functions need to be operated by the shop that is possible using the database. Such as storing customer information, products description, bills description etc. On the other hand customers can gather informations what they need from the shop. Such as info about products, shoppings, bill etc. So I think this database can support the business succesfully.

Main er diagram



Normalization :

1 . Here no multi value . so, does not need 1nf.

2.Here has no composite key . so , does not need 2nf.

3. In employee table there is a transitive dependency on manager rank and commission .

(before normalization )

Manager:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Manager id | Employee id | Rank | Commission | Educational information |

(After normalization )

manager

|  |  |  |  |
| --- | --- | --- | --- |
| Manager id | Employee id | Rank | Educational information |

Bonus:

|  |  |
| --- | --- |
| rank | commission |

Schema Diagram After Normalization :

There are 18 Entity table and relationship tables to relate the tables.

1.shop

CREATE TABLE SHOP (

SHOP\_ID VARCHAR2(13)PRIMARY KEY,

SHOP\_NAME VARCHAR2(100),

ESTABLISHED\_DATE DATE,

OWNER\_NAME VARCHAR2(100),

WEB\_ADDRESS VARCHAR2(50),

HEAD\_OFFICE\_ADDRESS VARCHAR2(200),

CONSTRAINTS SHOP\_ID\_CK CHECK (SHOP\_ID BETWEEN '111-0000' AND '111-9999')

)

2.

CREATE TABLE BRANCH (

BRANCH\_ID VARCHAR2(13)PRIMARY KEY,

SHOP\_ID VARCHAR2(13),

BRANCH\_NAME VARCHAR2(100),

BRANCH\_ADDRESS VARCHAR2(200),

BRANCH\_ESTABLISHED\_DATE DATE,

BRANCH\_EMAIL\_ID VARCHAR2(60),

BRANCH\_PHONE VARCHAR2(11),

CONSTRAINTS PHONE\_CK CHECK( BRANCH\_PHONE BETWEEN 01000000000 AND 01999999999 ),

CONSTRAINTS BRANCH\_ID\_CK CHECK (BRANCH\_ID BETWEEN '121-0000' AND '121-9999'),

FOREIGN KEY(SHOP\_ID) REFERENCES SHOP(SHOP\_ID)

)

3.

CREATE TABLE VEHICLE (

VEHICLE\_ID VARCHAR2(13) PRIMARY KEY,

BRANCH\_ID VARCHAR2(13),

VEHICLE\_NAME VARCHAR2(30),

VEHICLE\_TYPE VARCHAR2(20),

FOREIGN KEY (BRANCH\_ID) REFERENCES BRANCH(BRANCH\_ID),

CONSTRAINTS VEHICLE\_ID\_CK CHECK ( VEHICLE\_ID BETWEEN '131-0000' AND '131-9999' )

)

4.

CREATE TABLE EMPLOYEE(

EMPLOYEE\_ID VARCHAR2(13) PRIMARY KEY,

BRANCH\_ID VARCHAR2(13),

EMPLOYEE\_NAME VARCHAR2(30),

EMPLOYEE\_PHONE VARCHAR2(11),

EMPLOYEE\_ADDRESS VARCHAR2(100),

EMPLOYEE\_BIRTH\_DATE DATE,

JOINING\_DATE DATE,

SALARY NUMBER DEFAULT (5000),

JOB\_TYPE VARCHAR2(50),

WORKING\_ENTRY\_TIME VARCHAR2(20),

WORKING\_TIME NUMBER,

FOREIGN KEY(BRANCH\_ID) REFERENCES BRANCH(BRANCH\_ID),

CONSTRAINTS EMPLOYEE\_PHONE\_CK CHECK( EMPLOYEE\_PHONE BETWEEN 01000000000 AND 01999999999 ),

CONSTRAINTS EMPLOYEE\_ID\_CK CHECK (EMPLOYEE\_ID BETWEEN '141-0000' AND '141-9999')

)

5.

CREATE TABLE SALESMAN(

SALESMAN\_ID VARCHAR2(10)PRIMARY KEY,

EMPLOYEE\_ID VARCHAR2(13),

CONSTRAINTS SALESMAN\_ID\_CK CHECK( SALESMAN\_ID BETWEEN 'SALE-0000' AND 'SALE-9999')

)

6.

CREATE TABLE PRODUCT (

PRODUCT\_ID NUMBER PRIMARY KEY,

BRANCH\_ID VARCHAR2(13),

PRODUCT\_NAME VARCHAR2(30),

COMPANY\_NAME VARCHAR2(20),

TYPE VARCHAR2(20),

MEASUREMENT\_UNIT VARCHAR2(20),

COST\_PER\_UNIT NUMBER,

OFFER NUMBER,

PRODUCT\_QUANTITY NUMBER ,

FOREIGN KEY (BRANCH\_ID) REFERENCES BRANCH(BRANCH\_ID)

)

7.

CREATE TABLE ELECTRONICS (

PRODUCT\_SERIAL\_ID NUMBER PRIMARY KEY,

WARRANTY\_DATE DATE,

MODEL VARCHAR2(35),

MANUFACTURER\_COUNTRY VARCHAR2(30)

)

8.

CREATE TABLE ELECTRIC\_PRODUCT(

PRODUCT\_SERIAL\_ID NUMBER,

PRODUCT\_ID NUMBER,

FOREIGN KEY (PRODUCT\_SERIAL\_ID) REFERENCES ELECTRONICS(PRODUCT\_SERIAL\_ID),

FOREIGN KEY (PRODUCT\_ID) REFERENCES PRODUCT(PRODUCT\_ID)

)

9.

CREATE TABLE CUSTOMER(

CUSTOMER\_ID VARCHAR2(13) PRIMARY KEY,

BRANCH\_ID VARCHAR2(13),

CUSTOMER\_NAME VARCHAR2(40),

CUSTOMER\_ADDRESS VARCHAR2(100),

CUSTOMER\_PHONE NUMBER(11),

FOREIGN KEY (BRANCH\_ID) REFERENCES BRANCH(BRANCH\_ID),

CONSTRAINTS CUSTOMER\_ID\_CK CHECK (CUSTOMER\_ID BETWEEN '151-000000000' AND '151-999999999')

)

10

CREATE TABLE HOME\_DELIVERY\_WORKER (

WORKER\_ID VARCHAR2(13) PRIMARY KEY,

EMPLOYEE\_ID VARCHAR2(13),

DELIVERING\_AREA VARCHAR2(30),

DELIVERY\_TARGET NUMBER,

VEHICLE\_ID VARCHAR2(13),

FOREIGN KEY (VEHICLE\_ID) REFERENCES VEHICLE(VEHICLE\_ID),

FOREIGN KEY (EMPLOYEE\_ID) REFERENCES EMPLOYEE(EMPLOYEE\_ID),

CONSTRAINTS WORKER\_ID\_CK CHECK (WORKER\_ID BETWEEN '161-0000' AND '161-9999')

)

11.

CREATE TABLE BONUS(

RANK NUMBER PRIMARY KEY,

COMMISSION NUMBER

)

12.

CREATE TABLE MANAGER(

MANAGER\_ID VARCHAR2(13) PRIMARY KEY,

EMPLOYEE\_ID VARCHAR2(13),

RANK NUMBER,

EDUCATION VARCHAR2(200),

FOREIGN KEY (RANK) REFERENCES BONUS(RANK)

)

13.

CREATE TABLE BILL(

BILLING\_ID NUMBER PRIMARY KEY,

MANAGER\_ID VARCHAR2(13),

CUSTOMER\_ID VARCHAR2(13),

BILLING\_DATE DATE,

FOREIGN KEY (MANAGER\_ID) REFERENCES MANAGER(MANAGER\_ID),

FOREIGN KEY (CUSTOMER\_ID) REFERENCES CUSTOMER(CUSTOMER\_ID)

)

14.

CREATE TABLE BILL\_COST(

PRODUCT\_ID NUMBER,

BILLING\_ID NUMBER,

FOREIGN KEY (BILLING\_ID) REFERENCES BILL(BILLING\_ID),

FOREIGN KEY (PRODUCT\_ID) REFERENCES PRODUCT(PRODUCT\_ID)

)

15.

CREATE TABLE CUSTOMER\_BILL(

CUSTOMER\_ID VARCHAR2(13),

BILLING\_ID NUMBER,

FOREIGN KEY (BILLING\_ID) REFERENCES BILL(BILLING\_ID),

FOREIGN KEY (CUSTOMER\_ID) REFERENCES CUSTOMER(CUSTOMER\_ID)

)

16.

CREATE TABLE FOOD(

FOOD\_SERIAL\_ID VERCHAR2(20) PRIMARY KEY,

MANUFACTURE\_DATE DATE,

EXPIRE\_DATE DATE

)

17.

CREATE TABLE MANAGER\_EMPLOYEE(

MANAGER\_ID VARCHAR2(13),

EMPLOYEE\_ID VARCHAR2(13),

FOREIGN KEY (MANAGER\_ID) REFERENCES MANAGER(MANAGER\_ID),

FOREIGN KEY (EMPLOYEE\_ID) REFERENCES EMPLOYEE(EMPLOYEE\_ID)

)

18.

CREATE TABLE FOOD\_PRODUCT(

FOOD\_SERIAL\_ID NUMBER,

PRODUCT\_ID NUMBER,

FOREIGN KEY (FOOD\_SERIAL\_ID) REFERENCES FOOD(FOOD\_SERIAL\_ID),

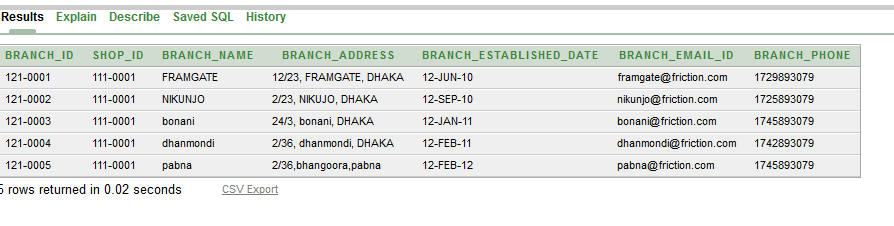
FOREIGN KEY (PRODUCT\_ID) REFERENCES PRODUCT(PRODUCT\_ID)

)

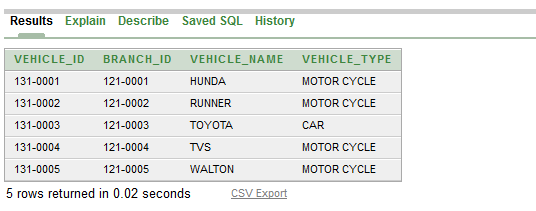
Values:

Shop: 

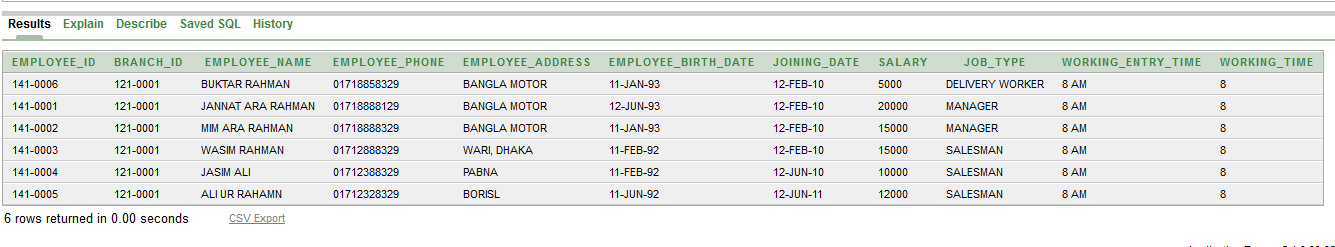
BRANCH:

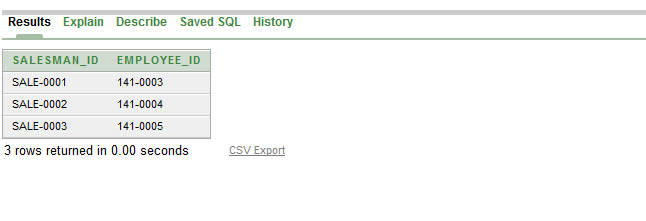


VEHICLE:

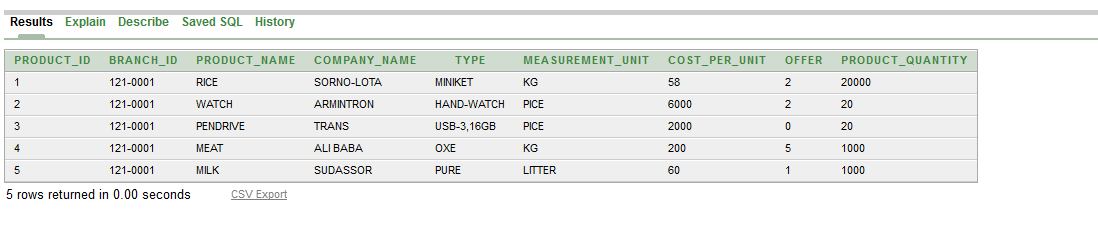


EMPLOYEE:

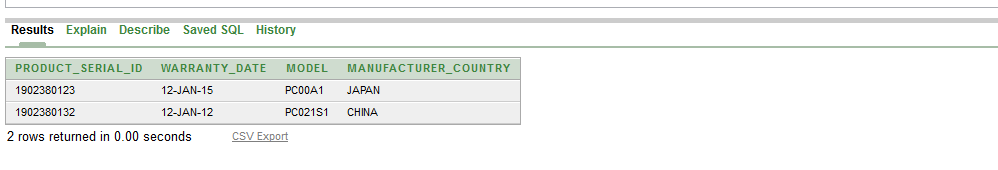


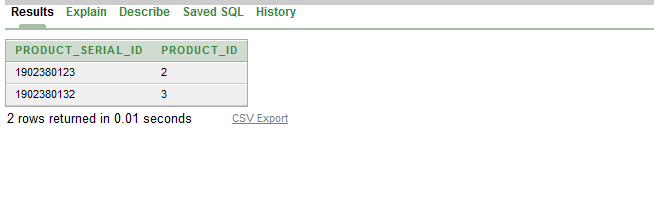
SALESMAN

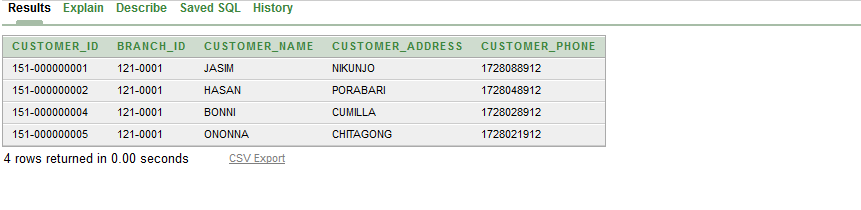
PRODUCT:



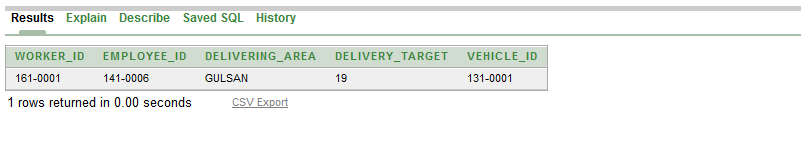
ELECTRONICS:

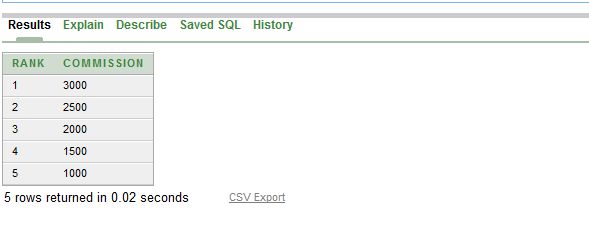


ELECTRIC\_PRODUCT

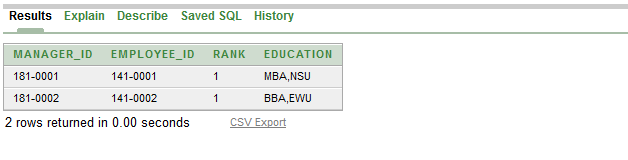
CUSTOMER:

HOME\_DELIVERY\_WORKER:

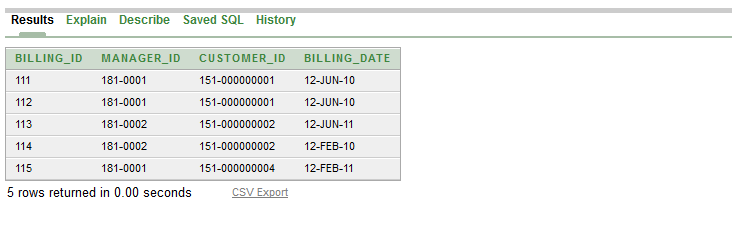


BONUS 

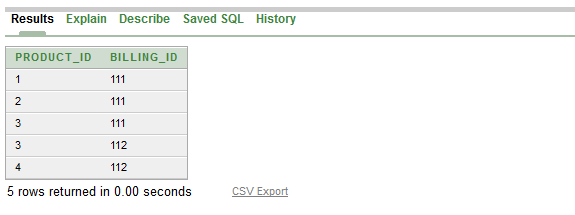
MANAGER



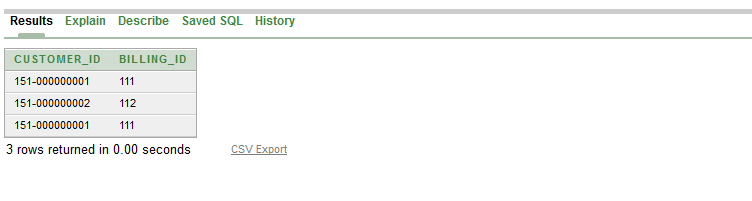
BILL:

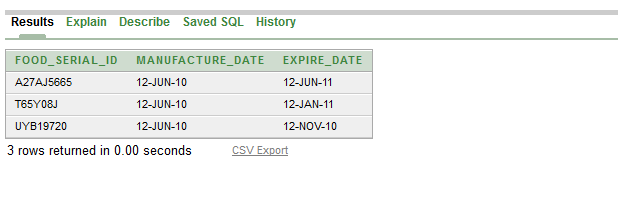


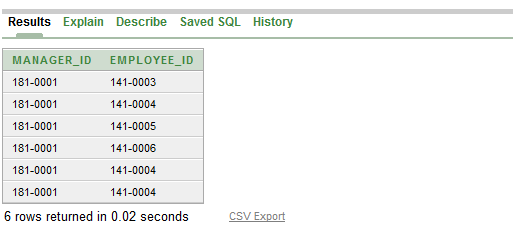
BILL\_COST:



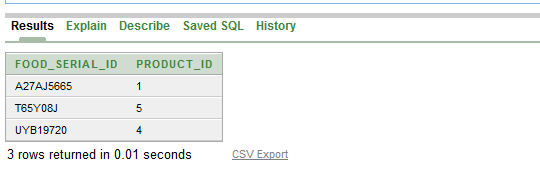
CUSTOMER\_BILL:



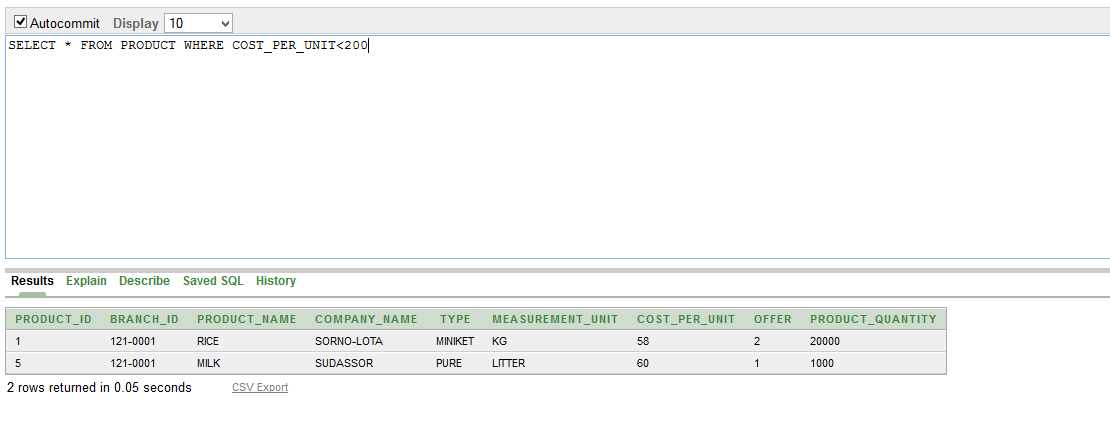
FOOD:

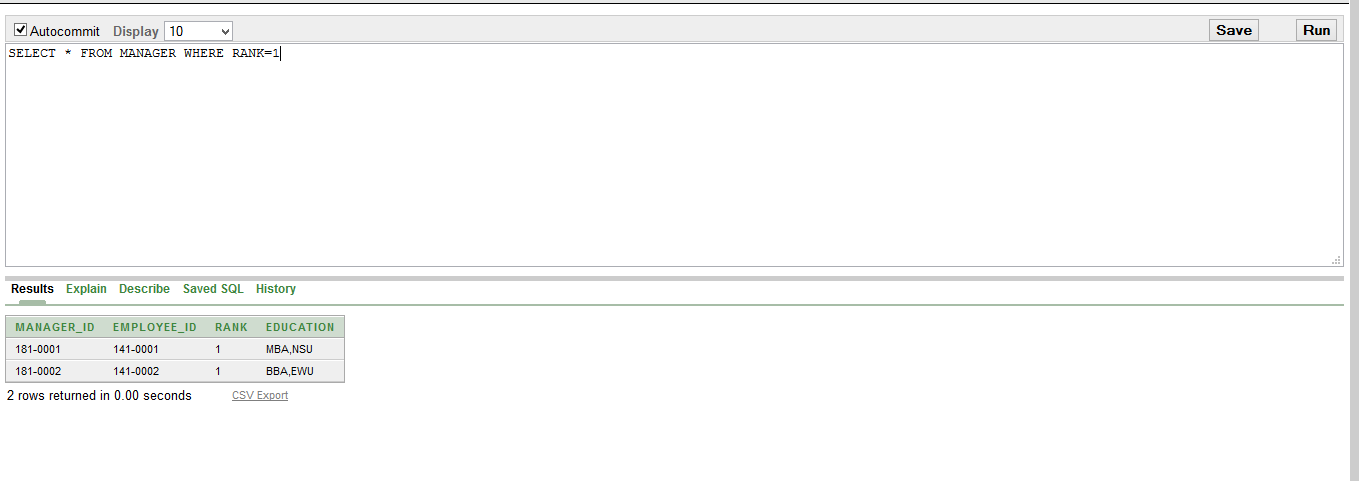
MANAGER\_EMPLOYEE:

FOOD\_PRODUCT:

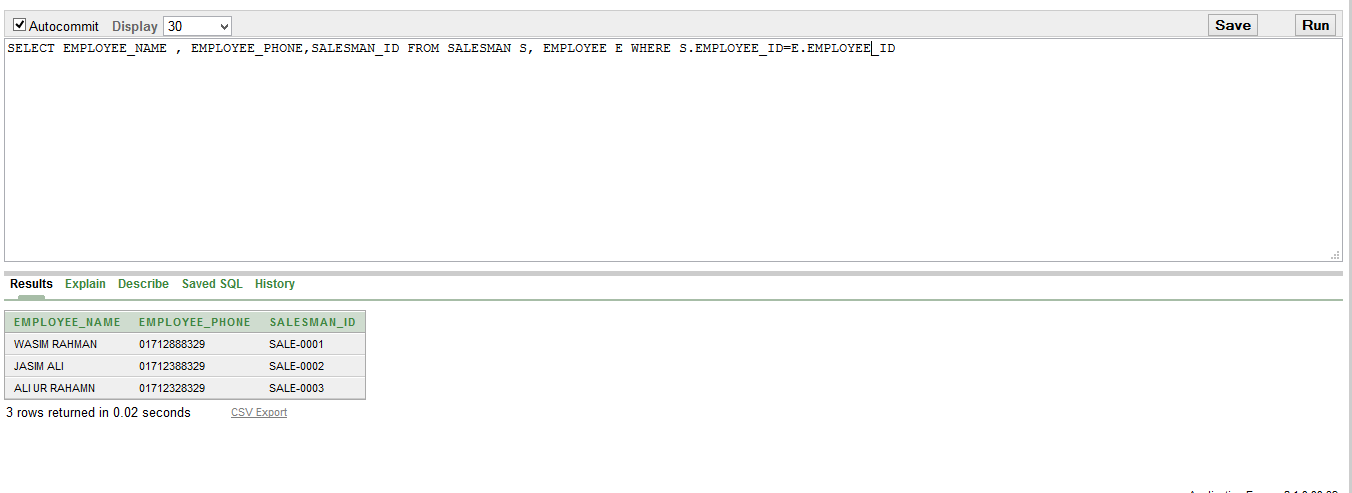


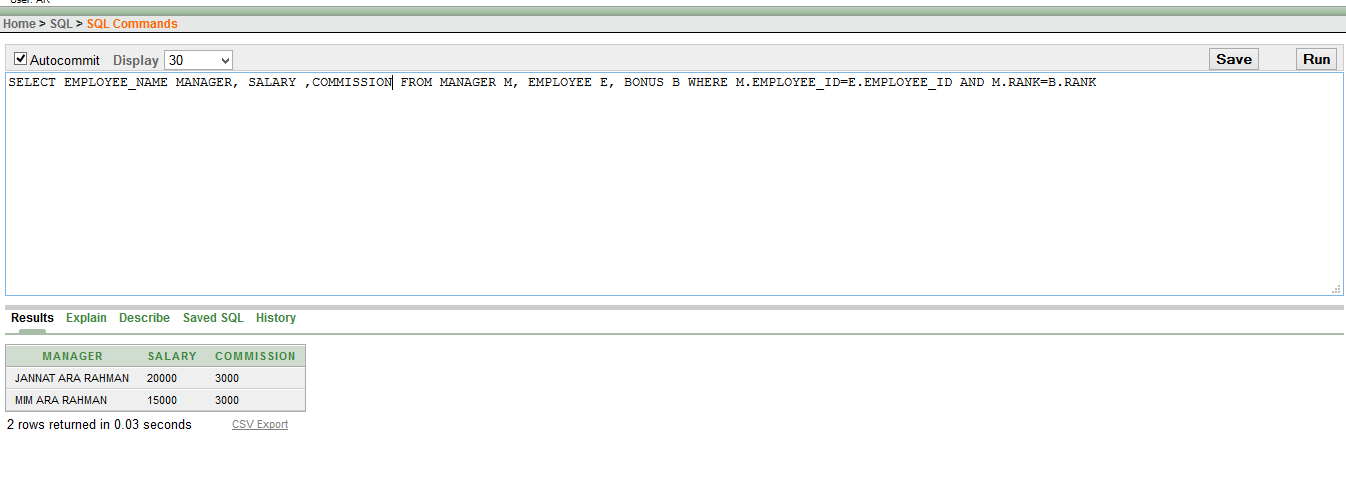
Some sql query and result:











---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Over all project

